CLAIMS

1. A method for developing, delivering and rendering a network-based computer application on a visual display connected to a network comprising the steps of:

developing a network-based application by a method comprising the steps of:

launching an integrated development environment that includes visual drag and drop capabilities designed to wire application components together;

using the development environment to define the structural and functional requirements of the network-based computer application;

using the visual drag and drop capabilities of the development environment to select at least one pre-built component capable of satisfying one of the requirements of the network-based application, each said component being written in a device independent computer scripting language;

causing the development environment to create a container document that represents the at least one selected pre-built component;

creating a bootstrap process document that may be used to initiate the network-based application, which bootstrap process document is written in a computer language that can be interpreted by a client device; and

deploying the network-based application on a computer that is connected to the network;

delivering the network application to a user by a method comprising the steps of:

storing the network-based application at a predetermined network address;

providing the bootstrap document from the network-based application to a user in response to initiation of a network communication session that identifies the pre-determined network address; and

causing the bootstrap process document to execute on the client device and thereby load the network-based application on the client device;

rendering the network-based application on the visual display of the client device by a method comprising the steps of:

retrieving at least a part of the network-based application into the client device during the network communication session;

causing the at least one pre-built component to be interpreted by the client device;

determining whether the interpreted component has a dependency that has not been satisfied;

if an unsatisfied dependency exists, deferring the interpretation of the component until all components have been loaded;

if no unsatisfied dependency exists, interpreting the component and creating an instance of the component on the client device;

upon completion of the interpretation of all components for which no unsatisfied dependency exists and loading of all components, reviewing each deferred component to determine if the component is an event;

if the deferred component is not an event, interpreting the component and creating an instance of the component on the client device;

if the deferred component is an event, registering the event on the client device in preparation for responding to a predetermined input or condition;

continuing to process components until all components have been instantiated and all events have been registered; and

creating a visual representation on the visual display.

2. The method of claim 1 wherein the pre-built components include at least one of:

a request broker;
a visual component;
a data component; or
a non-visual element.

3. The method of claim 1 wherein the bootstrap process document defines a standalone bootstrap process.

4. The method of claim 1 wherein the bootstrap process document defines a sibling bootstrap process.

5. The method of claim 1 wherein the bootstrap process document defines a dependent bootstrap process.

6. The method of claim 1 wherein the development environment uses a web face markup language.

7. The method of claim 1 further comprising the step of obfuscating at least one identifier prior to delivering the network-based application.

8. A method for developing, delivering and rendering a network-based computer application on a visual display connected to a network comprising the steps of:

developing a network-based application by a method comprising the steps of:

launching an integrated development environment that includes visual drag and drop capabilities designed to wire application components together;

using the development environment to define the structural and functional requirements of the network-based computer application;

using the visual drag and drop capabilities of the development environment to select at least one pre-built component capable of satisfying one of the requirements of the network-based application, each said component being written in a device independent computer scripting language;

causing the development environment to create a container document that represents the at least one selected pre-built component; and

deploying the network-based application on a computer that is connected to the network.

9. The method of claim 8 further comprising the step of creating a bootstrap process document that may be used to initiate the network-based application, which bootstrap process document is written in a computer language that can be interpreted by a client device.

10. The method of claim 8 further comprising the steps of:

storing the network-based application at a predetermined network address;

providing a bootstrap document from the network-based application to a user in response to initiation of a network communication session that identifies the pre-determined network address; and

causing the bootstrap process document to execute on the client device and thereby load the network-based application on the client device.

11. The method of claim 8 further comprising the steps of:

retrieving at least a part of the network-based application into the client device during the network communication session; and

causing the at least one pre-built component to be interpreted by the client device.

12. The method of claim 11 further comprising the steps of:

determining whether the interpreted component has a dependency that has not been satisfied;

if an unsatisfied dependency exists, deferring the interpretation of the component until all components have been loaded;

if no unsatisfied dependency exists, interpreting the component and creating an instance of the component on the client device;

upon completion of the interpretation of all components for which no unsatisfied dependency exists and loading of all components, reviewing each deferred component to determine if the component is an event;

if the deferred component is not an event, interpreting the component and creating an instance of the component on the client device;

if the deferred component is an event, registering the event on the client device in preparation for responding to a predetermined input or condition;

continuing to process components until all components have been instantiated and all events have been registered; and

creating a visual representation on the visual display.

13. The method of claim 8 wherein the pre-built components include at least one of:

a request broker;

a visual component;

a data component; or

a non-visual element.

14. The method of claim 10 wherein the bootstrap process document defines a standalone bootstrap process.

15. The method of claim 10 wherein the bootstrap process document defines a sibling bootstrap process.

16. The method of claim 10 wherein the bootstrap process document. defines a dependent bootstrap process.

17. The method of claim 10 further comprising the step of obfuscating at least one identifier prior to delivering the network-based application.

18. A method for developing, delivering and rendering a network-based computer application on a visual display connected to a network comprising the steps of:

storing the network-based application at a predetermined network address;

providing a bootstrap process document from the network-based application to a user in response to initiation of a network communication session that identifies the pre-determined network address; and

causing the bootstrap process document to execute on a client device and thereby load the network-based application on the client device.

19. The method of claim 18 further comprising the steps of:

launching an integrated development environment that includes visual drag and drop capabilities designed to wire application components together;

using the development environment to define the structural and functional requirements of the network-based computer application;

using the visual drag and drop capabilities of the development environment to select at least one pre-built component capable of satisfying one of the requirements of the network-based application, each said component being written in a device independent computer scripting language;

causing the development environment to create a container document that represents the at least one selected pre-built component; and

deploying the network-based application on a computer that is connected to the network.

20. The method of claim 19 further comprising the step of creating the bootstrap process document that may be used to initiate the network-based application, which bootstrap process document is written in a computer language that can be interpreted by the client device.

21. The method of claim 19 further comprising the steps of:

retrieving at least a part of the network-based application into the client device during the network communication session; and

causing at least one pre-built component to be interpreted by the client device.

22. The method of claim 21 further comprising the steps of:

determining whether the interpreted component has a dependency that has not been satisfied;

if an unsatisfied dependency exists, deferring the interpretation of the component until all components have been loaded;

if no unsatisfied dependency exists, interpreting the component and creating an instance of the component on the client device;

upon completion of the interpretation of all components for which no unsatisfied dependency exists and loading of all components, reviewing each deferred component to determine if the component is an event;

if the deferred component is not an event, interpreting the component and creating an instance of the component on the client device;

if the deferred component is an event, registering the event on the client device in preparation for responding to a predetermined input or condition;

continuing to process components until all components have been instantiated and all events have been registered; and

creating a visual representation on the visual display.

23. The method of claim 18 wherein the bootstrap process document defines a standalone bootstrap process.

24. The method of claim 18 wherein the bootstrap process document defines a sibling bootstrap process.

25. The method of claim 18 wherein the bootstrap process document defines a dependent bootstrap process.

26. The method of claim 18 wherein the development environment uses a web face markup language.

27. The method of claim 18 further comprising the step of obfuscating at least one identifier prior to delivering the network-based application.

28. A method for developing, delivering and rendering a network-based computer application on a visual display connected to a network comprising the steps of:

retrieving at least a part of a network-based application into a client device during the network communication session; and

causing at least one pre-built component to be interpreted by the client device.

29. The method of claim 28 further comprising the steps of:

determining whether the interpreted component has a dependency that has not been satisfied;

if an unsatisfied dependency exists, deferring the interpretation of the component until all components have been loaded;

if no unsatisfied dependency exists, interpreting the component and creating an instance of the component on the client device;

upon completion of the interpretation of all components for which no unsatisfied dependency exists and loading of all components, reviewing each deferred component to determine if the component is an event;

if the deferred component is not an event, interpreting the component and creating an instance of the component on the client device;

if the deferred component is an event, registering the event on the client device in preparation for responding to a predetermined input or condition;

continuing to process components until all components have been instantiated and all events have been registered; and

creating a visual representation on the visual display.